III. REMARKS

1. Claims 1, 2, 4-7, 9-16, 18-24, 26-30, 32-38, 40-43, 45-47, 49-58, 61, and 62 remain in the application. Claims 3, 8, 17, 25, 31, 39, 44, 48, 59, 60, and 63 have been cancelled without prejudice. Claims 1, 10, 12, 15, 29, and 43 have been amended. Support for the amendments may be found in the published application, for example, paragraphs [0008], [0038], [0039], [0042], and Figures 3 and 4.

2. Applicants respectfully submit that claims 1, 9, 12, 14-15, 23, 26, 28, 29, 37, 40, 42, 43, 46, 54, 56, 58, 61, and 62 are patentable over the combination of Meppelink et al. (US 5,542,069, "Meppelink"), Sullivan et al. (US 5,737,557, "Sullivan") and Balsara et al. (US 6,065,012, "Balsara") under 35 USC 103(a).

The combination of Meppelink, Sullivan and Balsara fails to disclose or suggest:

> launching a first view in said first target application by calling a method associated with said first target application by said view router, said launching comprising a presentation of a first user interface form in said computer graphical user interface by said first target application;
>
> receiving data to the first view from a user during said launching of the first view;
>
> continuing said view router by calling a listener method associated with said view router by said first target application;
>
> launching a second view in said second target application by calling a method associated with said second target application by said view router when entries for views not launched remain in said view chain, said launching comprising a presentation of a second user interface form in said computer graphical user interface by said second target application;
>
> receiving data to the second view from the user during said launching of the second view;
>
> continuing said view router by calling a listener method associated with said view router by said second target application; and

continuing said first application automatically when no entries for views not launched remain in said view chain data structure by calling a listener method associated with said first application by said view router,

as recited by amended claims 1, 15, 29 and 43.

Meppelink (US 5,542,069) discloses an input device emulator that comprises an interpreter for a windowed environment. The interpreter reads and executes commands from a file. The interpreter commands correspond to a variety of user interface control device actions and relate to the manipulation of the windowed environment and the entry of text via a keyboard. The interpreter simulates an actual user for mass-testing purposes. Meppelink also discloses that the windowed environment comprises an application program that communicates with a window server program. The window server program constructs window user interface elements and represents information obtained from the application program in these window user interface elements. The window server controls the display using lower level user interface display instructions. Normally, input from a user via an I/O device is processed in the window server program to map the user input, for example, mouse clicking at specified locations on the display to user interface events raised to the application program such as the selection of a menu item. However, in order to support mass-testing of user interfaces, an input device emulator is coupled to the window server to simulate user actions. The user actions are read from a test script. The benefit of placing a user interface emulator between the stream modules and the window server is that the fact whether a user interface action indication originates from an actual user or from an emulator script may be hidden from the window server. Thus, software testing may be made easier. Meppelink fails to disclose the use of the method in production mode, that is, in normal usage.

Sullivan (US 5,737,557) discloses a windowed environment where a number of collective actions may be associated with the files contained in a software suite. A software suite is a file and program folder displayed to the user as an icon that may be expanded to a window by user selection (Sullivan, column 5, row 58 – 60). It comprises files from different file system directories. Each software suite has associated with it a storage element in which is stored contextual information for defining the appearance and behaviour of the software suite window. A spot interface is designed to float on top of the desktop and all windows that are currently open. The placement of the mouse pointer symbol over the spot interface results in a

display of a circular visual element that encircles the spot interface. The circular visual element, in other words, a collar, comprises four quadrants each of which correspond to a secondary user interface. The selection of a quadrant in the collar results in the display of a secondary user interface that comprises a number of icons on the secondary user interface that are used to start an application or to open a file when the icon is clicked. To summarize, Sullivan discloses the mere fact that a program or a file may be started or opened by clicking a mouse button while the mouse cursor is over the corresponding icon.

Balsara (US 6,065,012) discloses a method for managing user-relevant data such as E-mail, calendar, contact and task list data. The user-relevant data is gathered from a multitude of applications. The method comprises the forming of a summary HTML page that consists of multiple separate tables. There is one table for each application. Each application to present its data in a table on the summary HTML page provides a specific method for gathering the data to be presented from the application. The summary HTML page source code comprises tags for defining each of the tables. In each table tag there is a reference to an object class that acts as a data source for the data to be fetched to the table rows and columns. For the object class are defined the parameter names corresponding to the table columns. The browser invokes a method in the data source object class upon encountering a table definition in the HTML source code. Upon having collected the data, the browser adds the HTML code for presenting the table data to the HTML page. The HTML page is finally presented by the browser to the user. In Balsara the table data is not presented to the display by the target application. The problem associated with this approach is that the view having a certain look-and-feel familiar to the user may not be reproduced by the browser, since the collected data is provided from the target application to the browser in raw format and must be presented using HTML to the user. More importantly Balsara fails to disclose the possibility for the user to add or edit data to a target application when a view, or rather a data collection method in the case of Balsara, is being executed in the target application. Balsara only discloses data collecting from a database or file while a table is being populated with data from a target application. In Balsara the editing is allowed via the browser only after the complete HTML form with all the tables from all the target applications has been presented. Balsara also fails to disclose the returning to an application that has determined a view route and provided it to a view router after the view route has been processed.

Reconsideration of the rejection is respectfully solicited in light of the following arguments:

As the Examiner admits in the final Office Action of 21 July 2009, Meppelink and Sullivan are silent on the features of determining a view chain data structure comprising at least three entries, each of said entries comprising an application identifier and a view identifier; a view identified by said view identifier is associated with an application identified by said application identifier; checking whether entries for views not launched remain in said view chain data structure, each said entry for a view not launched specifying a view identifier for a view not yet launched by said view router; launching a second view based on a second entry in said view chain data structure automatically by said view router when entries for views not launched remain in said view chain; and continuing said first application when no entries for views not launched remain in said view chain data structure.

The applicant respectfully submits that Balsara fails to disclose the features of the present claims missing from the combination of Meppelink and Sullivan, that is, *launching a first view in said first target application by calling a method associated with said first target application by said view router, said launching comprising a presentation of a first user interface form in said computer graphical user interface by said first target application; receiving data to the first view from a user during said launching of the first view; continuing said view router by calling a listener method associated with said view router by said first target application; launching a second view in said second target application by calling a method associated with said second target application by said view router when entries for views not launched remain in said view chain, said launching comprising a presentation of a second user interface form in said computer graphical user interface by said second target application; receiving data to the second view from the user during said launching of the second view; continuing said view router by calling a listener method associated with said view router by said second target application;* and *continuing said first application automatically when no entries for views not launched remain in said view chain data structure by calling a listener method by said view router* as claimed in amended claims 1, 15, 29 and 43.

2.1 Balsara fails to disclose or suggest launching a first view in said first target application by calling a method associated with said first target application by said view router, said launching comprising a presentation of a first user interface form in said computer graphical user interface by said first target application. In Balsara the execution of a data collection method provided by a target application from the HTML browser that is in the process of rendering the HTML document to the browser window does not involve the

presentation of a user interface form in a computer graphical user interface by the target application. In Balsara the collected data is presented at a later stage by the browser on the basis of an HTML table filled using the collected data. The HTML table is also filled only in the browser. No presentation of data occurs during the execution of the data collection method in Balsara.

2.2 Balsara also fails to disclose or suggest the feature of receiving data to the first view from a user during said launching of the first view. Balsara fails to disclose the possibility for the user to add or edit data to a target application when a view, which is rather a data collection method in the case of Balsara, is being executed in the target application. Balsara merely discloses data collecting from a database or a file while a table is being populated with data from a target application. In Balsara the editing of application data, that is, any user-relevant data, is allowed via the browser only after the complete HTML form with all the tables from all the target applications has been presented. This occurs so that the user selects an item on the complete HTML page by clicking the mouse. The user may input the changed data on the complete HTML page, which results in the changing of the data via a separate updating method associated with the application with which the data was related. Therefore, a person skilled in the art appreciates that the teaching in Balsara is very different. In Balsara the receiving of data to a view or even receiving of any data from the user is not possible during the launching of the first view.

2.3 Balsara also fails to disclose the feature of continuing said first application when no entries for views not launched remain in said view chain data structure by calling a listener method by said view router. In Balsara there is no disclosure of continuing an application different from the view router when the view chain has been processed. In Balsara after the rendering of the HTML page in the browser there is no application to be continued automatically by the browser. There is no application from which execution would have digressed to the view router and to which the execution must return. On the other hand, also Meppelink fails to disclose this feature, because the interpreter itself is in charge of processing the simulation file and after processing the simulation file there is no application to which return automatically.

For all the foregoing reasons, it is respectfully submitted that the amended independent claims 1, 15, 29 and 43 are patentable over Meppelink, Sullivan and Balsara. It is further submitted

that one skilled in the art would not have arrived at the solution as claimed in claims 1, 15, 29 and 43 in light of the teachings in Meppelink, Sullivan and Balsara. Because each of the independent claims of the present application are believed to be distinguished over the cited art, it is respectfully submitted that all the pending dependent claims which depend from one of the independent claims 1, 15, 29 and 43, directly or indirectly are further distinguished over the cited art. Thus, the combination of Meppelink, Sullivan and Balsara fails to render claims 1, 9, 12, 14-15, 23, 26, 28-29, 37, 40, 42-43, 46, 54, 56, 58 and 61-62 unpatentable.

3. Applicants respectfully submit that claims 2, 4-8, 10, 13, 16, 18-22, 24, 27, 30, 32-36, 38, 41, 45, 47, 49-53, 55, and 57 are patentable over the combination of Meppelink, Sullivan, Balsara and Bahrs et al. (US 7,181,686, "Bahrs") under 35 USC 103(a).

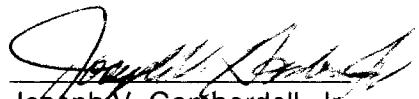
Claims 2, 4-8, 10, 13, 16, 18-22, 24, 27, 30, 32-36, 38, 41, 45, 47, 49-53, 55, and 57 each depend from one of independent claims 1, 15, 29 and 43.

Bahrs fails to disclose or suggest the features of the independent claims missing from the combination of Meppelink, Sullivan, and Balsara. Therefore, the combination of Meppelink, Sullivan, Balsara and Bahrs fails to render claims , 4-8, 10, 13, 16, 18-22, 24, 27, 30, 32-36, 38, 41, 45, 47, 49-53, 55, and 57 unpatentable.

For all of the foregoing reasons, it is respectfully submitted that all of the claims now present in the application are clearly novel and patentable over the prior art of record, and are in proper form for allowance. Accordingly, favorable reconsideration and allowance is respectfully requested. Should any unresolved issues remain, the Examiner is invited to call Applicants' attorney at the telephone number indicated below.

The Commissioner is hereby authorized to charge payment for any fees associated with this communication or credit any over payment to Deposit Account No. 16-1350.

Respectfully submitted,



21 October 2009

Joseph V. Gamberdell, Jr .Date
Reg. No. 44,695

Perman & Green, LLP
99 Hawley Lane
Stratford, CT 06614
(203) 259-1800
Customer No.: 2512